

他個体を参照した進化的計算による 巡回セールスマン問題の解法

佐藤 豊浩^{1,a)} 穴田 一^{1,b)}

受付日 2018年5月23日, 採録日 2018年12月4日

概要: 進化的計算は, 生物学的進化から着想を得た操作により, 生物個体を模した複数の解候補を解空間上で移動させることで高精度な解を探索するという近似解法の枠組みの1つである. しかし, 進化的計算の考えを基に設計された既存手法には, 自然界の生物の仕組みに囚われ過ぎている部分がある. 加えて, 探索集団が持ち合わせない新規要素の発見が乱数に強く依存している. そこで, 本研究では進化的計算の考えを基に, 自然界の仕組みよりも巡回セールスマン問題を解くことに重点を置いた, 乱数に強く依存しない突然変異を導入した新たなアルゴリズム, 参照進化 (Referential Evolution, RE) を構築した. そして, 巡回セールスマン問題のベンチマーク問題を用いて, 近似解を高精度かつ高速に求めることでその有効性を確認した.

キーワード: 進化的計算, 進化的アルゴリズム, 遺伝的アルゴリズム, 巡回セールスマン問題

A Novel Evolutionary Algorithm for Solving the Traveling Salesman Problem

TOYOHIRO SATO^{1,a)} HAJIME ANADA^{1,b)}

Received: May 23, 2018, Accepted: December 4, 2018

Abstract: Evolutionary computation (EC) is an evolution-inspired algorithm and is one of the approximation algorithms. EC's emphasis lies in the mechanism of biological evolution. Therefore, we constructed a new algorithm named Referential Evolution (RE) whose main emphasis is not biological evolution. This was done to solve non-deterministic polynomial-time hard combinatorial problems. We confirmed the effectiveness of our proposed algorithm by comparing it to other EC-based algorithms using several benchmark problems taken from the TSPLIB, which is a library of traveling salesman problem.

Keywords: evolutionary computation (EC), evolutionary algorithm (EA), genetic algorithm (GA), traveling salesman problem (TSP)

1. はじめに

本研究では, 進化的計算 (Evolutionary Computation, EC) の考えに基づいて巡回セールスマン問題 (Traveling Salesman Problem, TSP) の近似解を高精度かつ高速に求める新たなアルゴリズム, 参照進化 (Referential Evolution, RE) を構築した. TSP は, 配送計画や基板穿孔などの現

実社会の問題に直結しているため, 重要性の高い問題とされている. しかし, 問題規模の増加にともなう解候補数の増加が指数的であるため, 厳密解を現実的な時間内に総当たりで求めることは困難である. そのため, 近似解を高精度かつ高速に求める方法が研究されている. EC は, 生物学的進化から着想を得た操作により, 生物個体を模した複数の解候補を解空間上で移動させることで高精度な解を探索するという近似解法の枠組みの1つである. EC の考えを基に設計された既存手法には, 遺伝的アルゴリズム (Genetic Algorithm, GA) [1], 粒子群最適化 (Particle Swarm Optimization, PSO) [2], アントコロニー最適化

¹ 東京都市大学大学院工学研究科
Graduate School of Engineering, Tokyo City University, Setagaya, Tokyo 158-8557, Japan

a) g1681805@gmail.com

b) h-anada@tcu.ac.jp

(Ant Colony Optimization, ACO) [3]などが存在し、様々な最適化問題に対して優れた性能を発揮している。しかし、これらの手法を TSP に適用する場合、それぞれ様々な工夫を必要とする。たとえば、GA の操作を TSP に適用するには、問題特有の情報である巡回経路を適切に遺伝させる必要があり、様々な交叉方法が提案されている [4], [5]。こうした既存手法には、自然界の生物の仕組みに囚われ過ぎていている部分がある。加えて、探索集団が持ち合わせない新規要素の発見が乱数に強く依存している。

そこで本研究では EC の考えを基に、自然界の仕組みよりも TSP を解くことに重点を置いた、乱数に強く依存しない突然変異を導入した新たなアルゴリズム、RE を構築した。そして TSPLIB [10] に掲載されているベンチマーク問題を用いてこの手法と既存手法の比較を行った。その結果、提案手法が TSP の近似解を高精度かつ高速に求められることを確認した。

2. 巡回セールスマン問題 (TSP)

TSP とは、複数の都市座標と都市間のコストが与えられたとき、すべての都市を 1 度ずつ訪問して最初の都市へ戻るコストの総和が最小となる巡回経路を求める組合せ最適化問題である。TSP は、配送計画や基板穿孔などの現実社会の問題に直結しているため、重要性の高い問題とされている。TSP の解候補である巡回経路は、都市数を n としたとき $(n-1)!/2$ 通り存在する。したがって、問題規模の増加にともなう解候補数の増加が指数的であるため、厳密解を現実的な時間内に総当たりで求めることは困難である。このような問題は NP 困難 (NP-hard) と呼ばれ、厳密解ではなく近似解を高精度かつ高速に求める方法が研究されている。

3. 進化的計算

3.1 進化的計算 (EC)

EC は、生物学的進化から着想を得た操作により、生物個体を模した複数の解候補を解空間上で移動させることで、高精度な解を探索するという近似解法の枠組みの 1 つである。EC は、良い解どうしには何らかの類似構造が存在するという近接最適性の原理 (Proximate Optimality Principle) [11] を仮定して探索の効率化を図っている。EC の考えを基に設計された既存手法には、生物の進化や遺伝をモデル化した GA、鳥や魚などの群れの動きをモデル化した PSO、蟻の採餌行動をモデル化した ACO などが存在し、様々な最適化問題に対して優れた性能を発揮している。

3.2 遺伝的アルゴリズム (GA)

GA [1] は、選択、交叉、突然変異、淘汰と呼ばれる生物の進化や遺伝をモデル化した操作により探索を行う手法である。GA は、探索集団から複数の親個体を選択して、親

の要素を組み合わせる交叉と要素の一部を変化させる突然変異により新しい子個体を生成する。そして、淘汰により個体数を一定に保つ。これらの操作を繰り返すことにより探索を行う手法である。また、選択や淘汰の際に高精度な解を優先するような選択圧をかけることで、子は高精度な解に共通する要素を多く含むようになる。したがって、探索集団は有望な探索領域に重点的に分布し、集中的な探索を行うことができる。このような GA の操作を TSP に適用するには、問題特有の情報である巡回経路を適切に遺伝させる必要があり、様々な交叉方法が研究されている。枝交換交叉 (Edge Exchange Crossover) [4] と呼ばれる交叉方法では、親が持つ巡回経路のみを用いて子を生成する。枝組み立て交叉 (Edge Assembly Crossover) [5] と呼ばれる交叉方法では、親が持つ巡回経路を組み合わせる複数の部分閉路を生成し、それらを結合することで子を生成する。こうした交叉により生成される子は親の要素を十分に引き継ぐことができる。しかし、一般的な GA には自然界の生物の仕組みに囚われ過ぎていている部分がある。それは、親として 2 個体を選択している点と選択した親を対等に扱う点である。親の個体数に関しては 3 個体以上の研究 [6], [7], [8], [9] も存在するが、それらの既存研究でも親を対等に扱っているため、淘汰の結果、探索集団は一部の優秀な個体に強く影響を受けている。そのため、解空間上で分散している探索集団の初期配置を急速に局所解に収束させてしまい、多様な解空間の探索が行えていない。また、突然変異は突然変異確率による乱数で選ばれた経路の入れ替えを繰り返すため、乱数への依存が強く、総当たりと同様の非効率な探索と考えられる。

4. 提案手法

4.1 参照進化 (RE)

本研究では、EC の考えを基に新たなアルゴリズム、RE を構築した。RE は、既存手法の突然変異に代わり、問題特有の情報を表す個体の表現型を利用して、探索集団が持ち合わせない新規要素を含む要素集合を探索集団より生成する。そして、既存手法の交叉に代わり、各個体は複数の他個体や要素集合を参照し、進化する自個体の要素を優先して引き継ぎながら、新たな個体に進化する。そのため、乱数への依存を抑えられた突然変異、自個体の要素を優先的に引き継ぎながら他個体の要素を取り入れる交叉という、自然界の生物の仕組みに囚われない新たな操作となっている。また、各個体を中心とした近傍探索が行われるため、探索集団の分散した初期配置を活用し、解空間を多様に探索することが可能となっている。

TSP における問題特有の情報は座標空間上の巡回経路であるため、探索集団内の複数の個体が持つ経路の位置関係を利用することで、探索集団が持ち合わせない経路の発見を行い、それらを部分経路集合という要素集合として用い

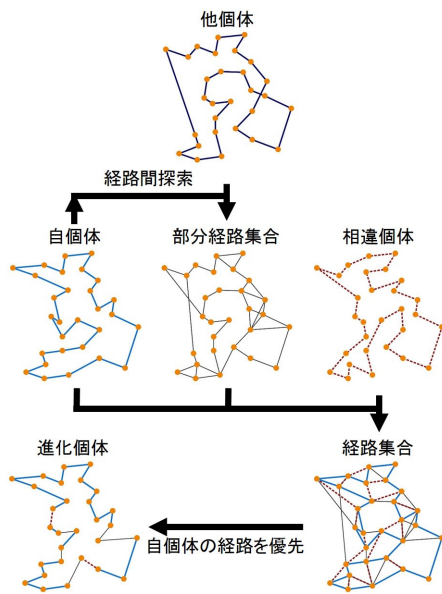


図 1 自個体を中心とした進化のイメージ
Fig. 1 Flow of evolution.

る。そして、各個体は複数の他個体の巡回経路と部分経路集合を参照し、新たな個体に進化する。その際、自個体の経路を優先して引き継ぐことで、各個体を中心とした近傍探索が行われる。参照する対象には、探索範囲を拡大させる要素と探索集団が持ち合わせない新規要素を含ませる。そのため、「自個体と解空間上で最も離れた相違個体（重複する経路数が最も少ない他個体）」と「部分経路集合」を参照の対象とする。部分経路集合は、進化する自個体と探索集団内の無作為に選択した他個体の表現型である巡回経路を利用して、互いの経路と経路の間を座標空間上で探索し、発見した経路より生成される。このとき、互いの経路を方向ベクトルとして扱うことで、互いの経路と類似した経路の発見を行う。この操作により生成された部分経路集合は探索集団が持ち合わせない経路を含む可能性があり、GA の突然変異で行われる経路の入れ替えと異なり、座標空間上の経路の位置関係を用いることで、乱数への依存を抑えて新規要素の発見を行うことができる。そして、自個体と相違個体と部分経路集合を重ね合わせた経路集合を用いて新たな個体に進化する。図 1 は提案手法により個体が進化する流れを表す。

4.2 RE の手順

RE は、以下の 4.2.1, 4.2.2 項の後 4.2.3~4.2.9 項を繰り返すことにより探索を行う。

4.2.1 初期探索集団の生成

すべての都市を 1 度ずつ通り、最初の都市に戻る巡回経路を表す個体をランダムに m 体 ($X^k, k = 1, \dots, m$) 生成する。 X_{ij}^k は個体 X^k が都市 i と都市 j 間の経路 ij を通る

か否かを 0, 1 で表し、通る場合 1 とする。ここで、 i と j は都市番号 ($i, j = 1, \dots, n$), n は都市数である。

4.2.2 評価

各個体の巡回経路を経路長により評価する。

4.2.3 終了判定

ステップ t が上限ステップ t_{max} に達したとき、探索集団から評価が最も良い個体を選び、解として出力してアルゴリズムを終了する。

4.2.4 選択個体 X^s の選択

探索集団から無作為に選択個体 X^s を選ぶ。

4.2.5 部分経路集合 V の生成

選択個体 X^s と探索集団内の他個体 X^r の巡回経路を利用して、互いの経路と経路の間を座標空間上で探索し、発見した経路より部分経路集合 V を生成する。このとき、経路を方向ベクトルとして扱い、共通した都市を始点とする方向ベクトルから合成ベクトルを求める。合成ベクトルは、始点と、互いの個体の方向ベクトルの終点を結ぶ経路を一定の比率で内分した点を結ぶベクトルで、互いの経路間の方向を表現する。そして、合成ベクトルと最も類似した経路に報酬を付与し、報酬が与えられた経路より部分経路集合 V が生成される。この操作により、問題特有の情報（座標空間上の巡回経路）を利用して、乱数への依存を抑えて新規要素の発見を行うことができる。以下の (1)~(3) に部分経路集合 V の生成の手順を示す。

(1) 他個体 X^r の選択

探索集団から無作為に他個体 X^r を選ぶ。

(2) 合成ベクトルの生成

選択個体 X^s と他個体 X^r の巡回経路のうち、ある都市 l を選択し、その都市を通る経路 $X_{al}^s, X_{lb}^s, X_{cl}^r, X_{ld}^r$ を利用する。ここで、都市 a と b は選択個体 X^s が都市 l の前後に訪問する都市、都市 c と d は他個体 X^r が都市 l の前後に訪問する都市である。各経路を都市 l が始点、もう一方の都市を終点とする方向ベクトルとして扱い、次式により合成ベクトル $\vec{v}_1 \sim \vec{v}_4$ を生成する。

$$\vec{v}_1 = F\vec{X}_{la}^s + (1 - F)\vec{X}_{lc}^r \quad (1)$$

$$\vec{v}_2 = F\vec{X}_{la}^s + (1 - F)\vec{X}_{ld}^r \quad (2)$$

$$\vec{v}_3 = F\vec{X}_{lb}^s + (1 - F)\vec{X}_{lc}^r \quad (3)$$

$$\vec{v}_4 = F\vec{X}_{lb}^s + (1 - F)\vec{X}_{ld}^r \quad (4)$$

ここで、 F は選択個体 X^s の比率を表すパラメータである。たとえば、 \vec{v}_1 は線分 ac を $(1 - F) : F$ に内分する、互いの経路間の方向を表現している。以下の図 2 は線分 ac を $(1 - F) : F$ に内分する \vec{v}_1 を生成するイメージを表す。

(3) 経路の決定と報酬の付与

合成ベクトルと最も類似した経路を決定して、経路に報酬を付与する。 \vec{v}_1 の終点に最も近い都市を都市 u とした場合、都市 l と都市 u 間の経路 lu の報酬 V_{lu} を 1 増加させる。同様に $\vec{v}_2 \sim \vec{v}_4$ より報酬の付与を行う。そして、都市

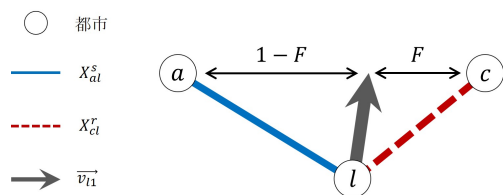


図 2 合成ベクトル生成のイメージ

Fig. 2 Image of creating resultant vector.

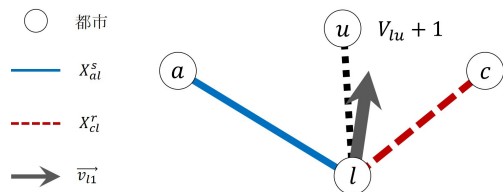


図 3 経路選択と報酬付与のイメージ

Fig. 3 Image of path selection and bonus.

l を他の都市に変更して、すべての都市についても同様に合成ベクトルの生成を行い、経路の選択と報酬の付与を行う。報酬が与えられた経路より部分経路集合 V が生成される。以下の図 3 は v_l の終点に最も近い都市 u と都市 l 間の経路の報酬 V_{lu} に報酬を付与するイメージを表す。

4.2.6 相違個体 X^d の選択

探索集団から選択個体 X^s と経路の重複が最も少ない相違個体 X^d を選ぶ。相違個体 X^d は選択個体 X^s と解空間上で最も離れた他個体であるため、参照することで探索範囲を拡大させる要素を効率良く取り入れることができる。

4.2.7 進化個体 E の構築

選択個体 X^s と部分経路集合 V と相違個体 X^d の経路を次式により重ね合わせ、経路集合 G を生成する。

$$G_{ij} = X_{ij}^s + \alpha V_{ij} + \beta X_{ij}^d \quad (5)$$

ここで、 α と β は進化個体 E を構築する際に部分経路集合 V と相違個体 X^d の経路を選択する優先度合いを表すパラメータである。このとき、 α と β を 1 未満に設定することで進化個体 E は選択個体 X^s の近傍探索を行うように構築される。以下の図 4 は選択個体 X^s と部分経路集合 V と相違個体 X^d の経路を重ね合わせ、経路集合 G を生成するイメージを表す。各経路の太さはそれぞれの経路の重みの大きさを表している。

そして、経路集合 G に含まれる経路から進化個体 E を構築する。初めに無作為に都市 i を選び、次に経路集合 G に含まれる経路を次式の選択確率により選択する。

$$P_{ij} = \frac{w_{ij}}{\sum_{h=1}^n w_{ih}} \quad (6)$$

$$w_{ij} = \frac{G_{ij}}{(d_{ij})^\gamma}$$

ここで、 d_{ij} は経路 ij の経路長、 γ は経路長の影響度合い

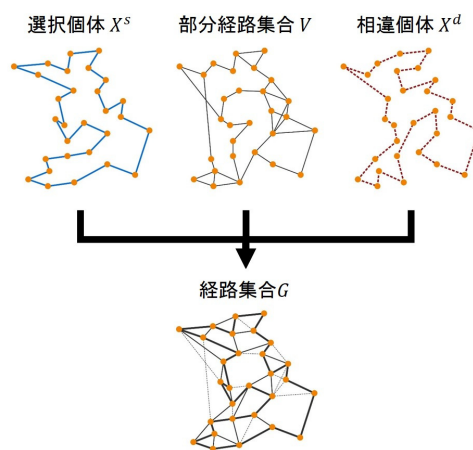


図 4 経路集合 G の生成のイメージ

Fig. 4 Image of generating path set.



図 5 2-opt 法による修正例

Fig. 5 An example of improvement using 2-opt.

を表すパラメータである。再び無作為に都市 i を選び直し、経路の選択を繰り返すことで巡回経路を構築する。このとき、新規要素を取り入れる部分経路集合 V と探索範囲を拡大させる相違個体 X^d が探索を多様化させる。また、経路集合 G に含まれる経路から選択不能な場合、例外処理として一時的に TSP の制約条件を満たすすべての経路の重み w_{ij} を次式により定義することで、経路集合 G に含まれない経路を選択する。

$$w_{ij} = \frac{1}{(d_{ij})^\gamma} \quad (7)$$

4.2.8 修正操作

構築した経路に対して 2opt 法による修正操作を行う。2opt 法は、巡回経路内の任意の 2 本の経路の入れ替えを繰り返すことでより短い巡回経路を構築する操作で、図 5 は 2opt 法によって巡回経路が改善された例である。左側の巡回経路に 2opt 法による修正操作を行うことで右下部分の経路が 2 本入れ替わり、右側の巡回経路が構築され、巡回経路が短くなっている。ほとんどのベンチマーク問題に対して、2opt 法のみで精度の良い近似解を求めることは不可能であるが、明らかに間違っている経路を容易に修正することができるため、厳密解への到達ステップや収束ステップの短縮化につながる。

4.2.9 更新

次式によって選択個体 X^s の更新を行う。

$$\begin{cases} E & \text{if } L^E < L^s * \min \left\{ 1 + \frac{diff}{n} * imp, lim \right\} \\ X^s & \text{Otherwise} \end{cases} \quad (8)$$

ここで、 L^s , L^E は選択個体 X^s , 進化個体 E の経路長, $diff$ は進化個体 E の経路の中で探索集団に含まれない経路の数, imp は $diff$ の影響度合い, lim は許容する改悪の倍率の上限を表すパラメータである. 探索集団が持ち合わせない経路を進化個体 E が発見したときのみ改悪が許容され, その経路の本数が多いほど改悪を許容する倍率が大きくなる. そして, $t \leftarrow t+1$ として 4.2.3 項へ戻る.

4.2.10 擬似コード

RE の手順の擬似コードを Algorithm 1 に示す. $\{\}$ で囲まれた部分はそのあとにどのような処理が行われるかを表し, $[\]$ で囲まれた部分は 4.2.1–4.2.9 項のどの部分かや, その行の処理に関する説明を表す.

5. 評価実験

5.1 既存手法との性能比較

TSPLIB [10] に掲載されているベンチマーク問題 (eil51, st70, kroA100) を用いて既存手法と性能比較を行い, 提案手法の有効性を確認した. 比較に用いた手法は, GA の交叉方法に枝交換交叉を用いた手法 [4] と ACO の 1 種である MAX-MIN Ant System に記憶領域を導入した手法 [12] である. これらの手法を選んだのは, 前者が GA を用いた TSP の手法で最もシンプルな手法であったためで, 後者は群知能アルゴリズムで最新と思われるものであったためである. 既存手法との性能比較にベンチマーク問題 (eil51, st70, kroA100) を用いたのは, 後者の手法で性能評価に用いていたためである.

条件を平等にするため, すべての手法の個体数は問題の都市数と同数, 初期探索集団の生成方法は 4.2.1 項と同様, ACO の記憶領域は近似アルゴリズムによる初期化を行わないものとした. 提案手法のパラメータは予備実験より様々な問題に対して解を高精度に求めることができた値, $F = 0.3$, $\alpha = 0.005$, $\beta = 0.1$, $\gamma = 2.0$, $imp = 1.0$, $lim = 1.001$ とし, $tmax$ は問題ごとに探索集団が十分に収束可能な値に決定した. その結果を表 1 に示す. 表 1 の opt は 200 試行中の厳密解到達回数を表し, error (%) は最良解の平均値が厳密解から離れている割合を表し, 次式で定義される.

$$error(\%) = \frac{L_{GB} - L_{opt}}{L_{opt}} \times 100$$

表 1 既存手法との性能比較
Table 1 Comparison experiment.

問題	GA		ACO		RE	
	opt	error(%)	opt	error(%)	opt	error(%)
eil51	21 / 200	0.912	74 / 200	0.2324	200 / 200	0
st70	18 / 200	1.0193	124 / 200	0.2637	200 / 200	0
kroA100	0 / 200	4.8823	132 / 200	0.0767	200 / 200	0

Algorithm 1 Referential Evolution の擬似コード

```

begin
  {初期探索集団の生成 (個体 m 体, n 都市) [4.2.1 項]}
  for(k = 0 to m - 1) do
    ランダムに都市 i を選択
    for(l = 0 to n - 2) do
      ランダムに都市 j を選択 (j ∈ N:未訪問都市集合)
       $X_{ij}^k = 1$ 
      i = j
    end for
    閉路作成を行う
  end for

  評価を行う [4.2.2 項]

  while t < tmax do [4.2.3 項]
    個体  $X^s$  選択 ( $X^s \in M$ :個体集合) [4.2.4 項]
    {部分経路集合 V の生成 [4.2.5 項]}
     $X^r \in M; (r \neq s)$  [(1) 無作為に他個体  $X^r$  選択]
    for(l = 0 to n - 1) do
       $\vec{v}_l = FX^s + (1 - F)X^r$  [(2) 合成ベクトルの生成]
      経路  $lu \approx$  都市 l +  $\vec{v}_l$  [(3) 経路決定]
       $V_{lu} ++$ ; [(3) 報酬付与]
    end for
    相違個体  $X^d$  選択を行う [4.2.6 項]
    ( $X^d \in M(d \neq s, \sum_i^n \sum_j^n (X_{ij}^s * X_{ij}^d)$  が最小))
    {進化個体構築 [4.2.7 項]}
     $G = X^s + \alpha V + \beta X^d$  [経路集合生成]
    ランダムに都市 i を選択
    for(l = 0 to n - 2) do
      if (G に選択可能経路が存在) then
        式 (6) により確率的に都市 j を選択
        (j ∈ G に含まれる未訪問都市集合)
      else
        式 (7) により確率的に都市 j を選択
      end if
       $E_{ij} = 1$ 
      i = j
    end for
    閉路作成を行う

    {修正操作 [4.2.8 項]}
    for(l = 0 to n - 2) do
      {2opt を行う}
      for(u = 0 to n - 1) do
        if(経路 la + 経路 ub > 経路 lu + 経路 ab) then 経路入替
          (入替前  $E_{la} = 1, E_{ub} = 1$ , 入替後  $E_{lu} = 1, E_{ab} = 1$ )
        end for
      end for

      {改悪許容更新 [4.2.9 項]}
      if (式 (8) の上側の条件を満たす) then
        更新を行う
      end if
      t ++
    end while

    最良解出力を行う
  end.

```

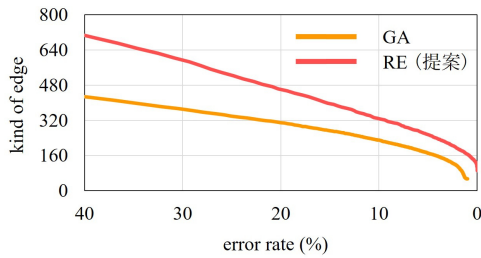


図 6 探索集団の経路種類数の比較
Fig. 6 Comparison of path kinds.

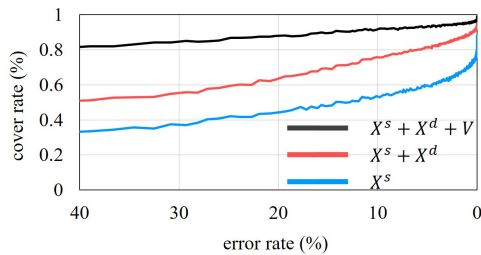


図 7 経路集合 G の cover rate
Fig. 7 Cover rate of path set.

ここで、 L_{GB} は求めた最良解 (Global Best) の長さ、 L_{opt} は厳密解の長さである。

表 1 の opt と error (%) より、すべての問題において提案手法は既存手法よりも高精度な解が求まることが確認された。これは、提案手法が各個体を中心とした近傍探索を行うことで、探索集団の多様性を維持しつつ探索が進行し、局所解に囚われなかったからであると考えられる。

そこで、そのことを確かめるために、error (%) に対する探索集団内の経路の種類数 (kinds of edge) を GA と比較した。提案手法と GA で eil51 を解いた際の error (%) に対する経路の種類数の 200 試行の平均を一定 step 刻みでプロットした結果を図 6 に示す。図 6 より提案手法は GA よりも探索集団内の経路の種類数が多く、多様な個体が保たれていることが確認された。

また、探索の進行具合に対して選択個体 X^s 、部分経路集合 V 、相違個体 X^d の経路を重ね合わせた経路集合 G の有効性を確認する。そのため、提案手法で eil51 を解いた際の error (%) に対する厳密解に含まれる経路を含んでいる割合 (cover rate) の 200 試行の平均を図 7 に示す。図 7 より、探索の序盤において青線で示された選択個体 X^s の cover rate は約 30% であり、黒線で示された選択個体 X^s と部分経路集合 V と相違個体 X^d を重ね合わせた経路集合 G の cover rate は約 80% である。このことより、相違個体 X^d や部分経路集合 V の生成は探索を多様化させることに有効であることが確認された。

加えて、提案手法と GA で eil51 を解いた際にかかる時間 (time) に対する error (%) の 200 試行の平均を図 8 に示す。図 8 よりすべての時間帯において赤線で示された提

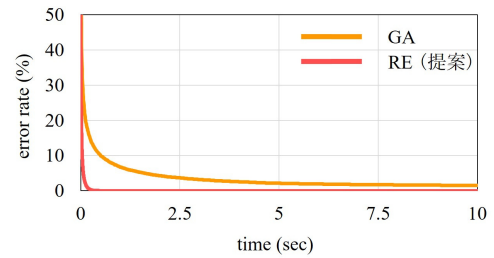


図 8 最良解の時間発展の比較
Fig. 8 Time evolution of solution.

案手法は黄線で示された GA よりも高精度な解が求められている。このことより、GA よりも高速に解を求めることができることが確認された。

5.2 ベンチマーク問題 (30 種類) に対する性能

5.1 節の比較実験によって提案手法が比較に用いた既存手法より優れた解探索能力を持つことが確認されたので、ここでは TSPLIB に掲載されているベンチマーク問題 (都市数 48~1,002 の問題から 30 問) を用いて、提案手法の性能を確認した。すべての問題において提案手法の個体数は 100 に固定し、それ以外のパラメータは 5.1 節の比較実験と同様である。rat575, rat783, pr1002 を 10 試行、それ以外の問題を 50 試行行った、改悪を認めなかった場合の結果を表 2 に、改悪を認めた場合の結果を表 3 に示す。これらの表の ave は解の平均値、opt は試行回数中の厳密解到達回数、error (%) は最良解の平均値が厳密解から離れている割合、step は各試行の最終更新 step の平均、 t_{max} は上限 step、表 3 の赤字の結果は改悪を認めた場合の方が良くなったことを表している。表 2 と表 3 の error (%) より、提案手法が求めた解は厳密解から 0.1% 以内に収まっており、高精度な解が求まること、問題の都市数だけではなく都市配置により解が求まる step が大きく変化することが確認された。また、更新の際に改悪を認めた方がほとんどの問題の場合において解の精度があがるか、解が求まるステップ数が早くなることが確認できた。

6. 考察

この章では新たに導入した操作の設計と役割を考察する。RE で突然変異の役割を果たす「部分経路集合 V の生成」は、選択個体 X^s と他個体 X^r が持つ巡回経路を利用して探索集団が持ち合わせない経路を発見する操作である。この操作により、探索序盤では選択個体 X^s と他個体 X^r の巡回経路の差が大きいため新規経路を発見しやすく、探索が進むにつれて選択個体 X^s と他個体 X^r の巡回経路の差が小さくなり、探索が収束していく。また、GA などで行われる遺伝型ではなく表現型である問題特有の情報 (座標空間上の巡回経路) を利用しているため、乱数に依存せずに新規経路の発見を行うことができる。この操作では、選

表 2 ベンチマーク問題に対する性能 (改悪を認めない場合)

Table 2 Experimental results for the TSPLIB problems without permitting peioration.

problem	ave	error (%)	opt	step	tmax
att48	33522.00	0.00000	50/50	1891.8	20000
eil51	426.00	0.00000	50/50	2238.4	20000
berlin52	7542.00	0.00000	50/50	758.9	20000
st70	675.00	0.00000	50/50	3429.9	100000
eil76	538.00	0.00000	50/50	3844.3	100000
pr76	108159.00	0.00000	50/50	3964.7	100000
rd100	7910.00	0.00000	50/50	4572.7	100000
kroA100	21282.00	0.00000	50/50	2064.2	100000
eil101	629.00	0.00000	50/50	11092.3	100000
lin105	14379.00	0.00000	50/50	1466.8	100000
ch130	6110.00	0.00000	50/50	21762.1	100000
ch150	6528.00	0.00000	50/50	5654.2	100000
kroA150	26524.00	0.00000	50/50	9588.3	100000
pr152	73682.00	0.00000	50/50	2587.2	100000
u159	42080.00	0.00000	50/50	2931.5	100000
rat195	2323.00	0.00000	50/50	27035.3	100000
d198	15780.00	0.00000	50/50	37074.7	100000
kroA200	29368.00	0.00000	50/50	26543.5	100000
tsp225	3916.00	0.00000	50/50	32759.5	100000
pr226	80369.00	0.00000	50/50	6573.3	100000
pr264	49135.00	0.00000	50/50	29680.7	100000
a280	2580.00	0.00000	50/50	17168.8	100000
pr299	48191.00	0.00000	50/50	103446.0	2000000
lin318	42029.00	0.00000	50/50	184230.4	2000000
rd400	15281.02	0.00013	49/50	1121696.6	5000000
pr439	107217.66	0.00062	49/50	213656.5	2000000
pcb442	50780.32	0.00457	49/50	868117.5	5000000
rat575	6773.30	0.00443	7/10	2631803.0	10000000
rat783	8806.20	0.00227	9/10	2101678.9	10000000
pr1002	259154.70	0.04235	4/10	8968756.7	10000000

表 3 ベンチマーク問題に対する性能 (改悪を認めた場合)

Table 3 Experimental results for the TSPLIB problems with permitting peioration.

problem	ave	error (%)	opt (%)	step	tmax
att48	33522.00	0.00000	50/50	1603.4	20000
eil51	426.00	0.00000	50/50	2245.0	20000
berlin52	7542.00	0.00000	50/50	722.5	20000
st70	675.00	0.00000	50/50	3605.2	100000
eil76	538.00	0.00000	50/50	3923.8	100000
pr76	108159.00	0.00000	50/50	3656.0	100000
rd100	7910.00	0.00000	50/50	4505.0	100000
kroA100	21282.00	0.00000	50/50	2198.5	100000
eil101	629.00	0.00000	50/50	10177.5	100000
lin105	14379.00	0.00000	50/50	1511.1	100000
ch130	6110.00	0.00000	50/50	20816.9	100000
ch150	6528.00	0.00000	50/50	5797.5	100000
kroA150	26524.00	0.00000	50/50	9000.1	100000
pr152	73682.00	0.00000	50/50	2567.8	100000
u159	42080.00	0.00000	50/50	2913.4	100000
rat195	2323.00	0.00000	50/50	25914.8	100000
d198	15780.00	0.00000	50/50	36305.2	100000
kroA200	29368.00	0.00000	50/50	25832.0	100000
tsp225	3916.00	0.00000	50/50	31626.1	100000
pr226	80369.00	0.00000	50/50	6673.3	100000
pr264	49135.00	0.00000	50/50	24545.1	100000
a280	2580.00	0.00000	50/50	16839.6	100000
pr299	48191.00	0.00000	50/50	102290.9	2000000
lin318	42029.00	0.00000	50/50	166822.2	2000000
rd400	15281.04	0.00026	48/50	823000.3	5000000
pr439	107217.00	0.00000	50/50	243585.4	2000000
pcb442	50778.00	0.00000	50/50	468885.5	5000000
rat575	6773.70	0.01034	3/10	2467614.4	10000000
rat783	8806.00	0.00000	10/10	1518407.0	10000000
pr1002	259067.70	0.00876	9/10	6978930.2	10000000

択個体 X^s と他個体 X^r の経路と経路の間をパラメータ F で設定した比率で探索する. $F = 0.5$ と設定した場合は最も広い範囲で新規経路の探索を行うが, 探索集団の収束速度が遅くなる. また, $F > 0.5$ と設定した場合は選択個体 X^s が優先されるため, 部分経路集合 V に新規経路が含まれにくくなる. よって, $F \leq 0.5$ が新規経路の発見と探索集団の収束速度のバランスがとれた値であると考えられる. そして, 予備実験より $F = 0.3$ が最もバランスの良い値であると明らかになった. RE で交叉の役割を果たす「経路集合 G の生成」は, 選択個体 X^s と部分経路集合 V と相違個体 X^d の経路を重ね合わせる操作である. この操作は, パラメータ α, β によって選択個体 X^s の経路を優先して引き継ぎ, 各個体を中心とした近傍探索が行われるように設計されている. パラメータ β は相違個体 X^d の経路の優先度合いを表し, この値が小さいほど選択個体 X^s の経路が引き継がれやすくなる. パラメータ α は部分経路集合 V の経路の優先度合いを表し, この値が大きいと新たな経路を取り入れやすくなる. しかし, 新たな経路を取り入れすぎると構築される巡回経路が大きく変化し, 近傍探索ではなくなる. また, 部分巡回経路には最終的に個体の評価を下げてしまう長い経路が含まれやすい. そこで, ヒューリ

スティックな情報である経路長をパラメータ γ で調整して取り入れている. よって, $\alpha \leq \beta < 1.0$ が新規経路の発見と探索集団の収束速度のバランスがとれた値であると考えられる. そして, 予備実験より $\alpha = 0.005, \beta = 0.1$ が最もバランスの良い値であると明らかになった. RE の「選択個体 X^s の更新」は, 改悪を許容することにより局所解から脱出する性能を向上させている. この操作は, 探索集団の多様性を確保するために設けている. そのため, 収束が遅くなるだけの改悪は起こらないように, 新規経路を含む割合が大きいときほど改悪を認めるように設計した.

本研究では相違個体, 自個体と無作為に選択した個体の経路の合成ベクトルによる新規経路の獲得, 自個体を優先した進化を導入した. 図 6 より相違個体や経路の合成ベクトルによる新規経路獲得が有効であること, 図 7 より自個体を優先した進化が経路の種類を保ちつつ進化することを可能にしていることが確認でき, これらのことが解探索に重要な役割を果たしていることが分かる.

7. むすび

本研究では, 進化的計算 (Evolutionary Computation, EC) の考えを基に巡回セールスマン問題 (Traveling Sales-

man Problem, TSP) の近似解を高精度かつ高速に求める新たなアルゴリズム, 参照進化 (Referential Evolution, RE) を構築した. 提案手法では, 乱数に依存しない突然変異, 個体数を任意に変更できる交叉, 探索集団の多様性を考慮した更新を既存の操作に代わり導入し, 解の精度向上を図った. そして, 既存手法との性能比較を行うことによりベンチマーク問題の高精度な解が高速に求まることを確認し, EC の課題である探索集団の収束性と多様性のバランスが適切に調整されたアルゴリズムを構築できたことを確認した.

今後の課題として, 本研究では「選択個体 X^s と他個体 X^r の巡回経路を利用して生成された部分経路集合 V 」と「解空間上で選択個体 X^s から最も離れた相違個体 X^d 」を参照したが, 提案手法は複数の他個体や要素集合を参照して扱うことができる拡張性を有するため, 参照する対象について検討余地がある. また, TSP 以外の最適化問題においても, 個体の表現型から新規要素を含む要素集合を生成する操作や, 複数の他個体や要素集合を参照することで新たな個体を生成する操作を, それぞれの最適化問題に応じて置き換えることで適用することができるため, RE の TSP 以外の最適化問題に対する有効性を確認したいと考えている.

参考文献

- [1] Holland, J.H.: *Adaptation in Natural and Artificial Systems*, The MIT Press, Commonwealth of Massachusetts (1975).
- [2] Kennedy, J. and Eberhart R.: Particle Swarm Optimization, *Proc. IEEE/ICNN*, pp.1942–1948 (1995).
- [3] Dorigo, M., Maniezzo, V. and Coloni, A.: Ant system: Optimization by a colony of cooperating agents, *IEEE Trans. Syst. Man. Cybern., Part B*, Vol.26, No.1, pp.29–41 (1996).
- [4] Maekawa, K., Tamaki, H., Kita, H. and Nishikawa, Y.: A method for the traveling salesman problem based on the genetic algorithm, *Trans. SICE*, Vol.31, No.5 (1995).
- [5] 永田裕一, 小林重信: 巡回セールスマン問題に対する交叉: 枝組み立て交叉の提案と評価, *人工知能学会誌*, Vol.14, No.5, pp.848–859 (1998).
- [6] Umbarkar, A.J. and Sheth, P.D.: Crossover Operators in Genetic Algorithms: A review, *ICTACT Journal on Soft Computing*, Vol.6, No.1, pp.1083–1092 (2015).
- [7] Zhihua, C. and Jianchao, Z.: Multi-parent Dynamic Nonlinear Crossover Operator for TSP, *IJCSNS International Journal of Computer Science and Network Security*, Vol.6, No.2, pp.103–106 (2006).
- [8] Ting, C.-K., Su, C.-H. and Lee, C.-N.: Multi-parent extension of partially mapped crossover for combinatorial optimization problems, *Expert Systems with Applications*, Vol.37, pp.1879–1886 (2010).
- [9] Ahmed, Z.H.: Multi-parent extension of sequential constructive crossover for the travelling salesman problem, *International Journal of Operational Research*, Vol.11, No.3, pp.331–342 (2011).
- [10] TSPLIB, available from (<http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>).

- [11] 柳浦睦憲, 茨木俊秀: 組合せ最適化: メタ戦略を中心として, 朝倉書店, 東京 (2001).
- [12] 磯崎敬志, 穴田 一: 記憶した最適解を参照する MAX-MIN Ant System による巡回セールスマン問題の解法, *電子情報通信学会論文誌*, Vol.J100-D, No.7, pp.1–9 (2017).



佐藤 豊浩

2018年東京都市大学大学院工学研究科システム情報工学専攻修了. 組合せ最適化問題の近似解法に関する研究に従事.



穴田 一 (正会員)

現在, 東京都市大学講師. 物理学会, 電子情報通信学会, 人工知能学会, 進化計算学会, 数理生物学会, 応用数理学会各会員. 複雑系の研究に従事.